The **World's #1 Computer** and **Videogame** Authority

PC CD-ROM ■ Nintendo 64 ■ Saturn ■ PlayStation ■ 3DO ■ Ar

EVERY
PlayStation, Nintendo 64, and Saturn game reviewed!

NEXT
GENERA N

350 GAMES REVIEWED
Don't buy a turkey this Christmas!

Leading edge computer and video **games**

January 1997

Want the full picture? Then turn the page...

Sony's New PlayStation

Coming in '97: The console that lets *you* make the games

EXCLUSIVE: How to make your own PlayStation games (and then how to sell them). **PLUS:** Tips from the pros

volume **three**

25

$7.99 U.S. $8.99 CAN.

01 >

0 71486 03676 0

REVEALED!
Everything they *don't* teach you at the DigiPen videogame school

Yarouze (the Japanese word roughly translates as "Let's Do It!") is a new program from Sony designed to enable anyone with a PC or Mac to make PlayStation games. It's a revolutionary concept, and Next Generation has the exclusive story on page 38

PARAPPA THE RAPPER: Yes! It's another one of those quirky Japanese games! Page 84

RESET
POWER
SONY
PlayStation

So could *you* make a game as good as *Nights*?

Cool gift, right? An exclusive version of **Yuji Naka's *Nights*** for Saturn — from Sega of America and **Next Generation**. We hope you enjoy this, your essential ***Next Generation Disc*** — and we also hope you enjoy this month's lead feature.

The sleek black PlayStation you see on these pages **hasn't been revealed in the U.S. before.** It is a **special edition** PlayStation designed to form the centerpiece of a **1997 program from Sony** called Yarouze **(the Japanese world roughly translates as "Let's Do It")**. **Gamers** who buy a Yarouze system (Sony estimates that it will sell for around $750) get a **black PlayStation**, a **serial link** to connect it your Mac or PC, **a CD of PlayStation programming libraries**, and membership to an **online resource** of tips, advice, and updates from Sony Computer Entertainment of America.

And what does all this add up to? All the equipment you need to make your own PlayStation games. This is a **revolutionary concept.** But it's **not a new one.**

In the **early 1980s**, young upstarts like **Dave Perry** and **Peter Molyneux** were **programming their first games** on **8-bit "home computers"** such as the Commodore 64 and the Sinclair Spectrum. These game machines (make no mistake, the **"home computer"** tag was usually just a marketing ploy) could be programmed **by anyone who chose to learn.** They came equipped with a keyboard, writable memory, and a chunky instruction manual. Even **game magazines of the time** were filled with program listings, and for **budding** game makers looking to learn their trade, it was the **perfect environment.** **But since that time** since that time, the low-cost home computers have been replaced by videogame consoles (with no keyboards or programming instruction manuals). Simultaneously, the **sophistication of PC technology** has sped beyond the grasp of ordinary gamers. For most gamers, "making your own game" would be practically impossible.

Until now. With a modest grasp of the **programming language C**, all gamers can start on the road to becoming a game developer. And who will be the **next Dave Perry or Peter Molyneux?** **It could be you...**

NEXT GENERATION

NEXT GENERATION

January 1997

Contact

Next Generation
Imagine Publishing, Inc.
150 North Hill Drive
Brisbane CA 94005
Advertising 415. 468. 4684
Editorial 415. 468. 4684
(E-mail: ngonline@imagine-inc.com)
FAX 415. 468. 4686

If you have questions about subscription, please contact:
Customer Service 415. 468. 4869
Customer Service fax 415. 656. 2486
(E-mail: subscribe@imagine-inc.com with "Next Generation" as the subject)

Editorial

Neil West editor-in-chief
Diane Anderson managing editor
Chris Charla still hanging around
Jeff Lundrigan reviews editor
Tom Russo assistant editor
Amos Parker editorial inspiration
Mike Wilmoth senior art director
Richard Szeto associate art director

Colin Campbell editor, *ng online*
Christian Svensson new media editor

Rick Sanchez disc editor
Jason Michaels associate disc editor

Editorial Contributors
Patrick Baggatta, Jason Bates, Roger Burchill, Nicolas di Costanza, Douglass Perry, Mike Salmon, Amy Tong, Todd Vaughn, Marcus Webb, Mike Wolf, Bernard Yee

Photography & Artistic Contributors
Mark Koehler, Aaron Lauer, Debbie Wells, Emil Yanos

Advertising

Doug Faust advertising manager
Aldo Parcheesi Royal game of India
Larae Brown marketing specialist
Kim Smith ad coordinator

Production

Richard Lesovoy production director

Imagine Publishing, Inc.

Jonathan Simpson-Bint publisher
Tom Hale director CD-ROM publishing
Bruce Eldridge newsstand director
Thea Selby newsstand manager
Terry Lawson newsstand analyst
Holly Klingel vp circulation
Gail Egbert circulation consultant
Kate Bailey circulation manager
Jane Jarvis fulfillment coordinator
Tom Valentino vice president/CFO
Chris Anderson president

Next Generation (ISSN# 1078-9693) is published monthly by Imagine Publishing, Inc, 150 North Hill Drive, Brisbane, CA 94005, USA. Periodicals Class postage paid at Brisbane, CA, and additional mailing offices. **POSTMASTER** send changes of address to Next Generation, PO Box 5368o, Boulder, CO 80322-3680. Newsstand distribution is handled by Curtis Circulation Company. Subscriptions: one year (12 issues) US: $34. Canada: $48. Foreign: $58. Canadian price includes postage and GST (GST# R 128 220688) CPC Int'l Pub Mail #0781126.

International Licensing Representative
Robert J. Abramson and Associates, Inc.
720 Post Road
Scarsdale, New York, 10583

Entire contents copyright 1997, Imagine Publishing, Inc. All rights reserved. Reproduction in whole or in part without permission is prohibited. Imagine Publishing, Inc. also publishes *Ultra Game Players, Boot, PC Gamer, MacAddict* and *The Net.* Products named in these pages are tradenames or trademarks of their respective companies. Imagine Publishing, Inc., is not affiliated with the companies or products covered in **Next Generation.**

Produced in the United States of America.

Standard Mail enclosed in the following editions: G, G1

Bulk Rate
U.S. Postage Paid
South Florida Facility
Permit No. 161

Next Generation shares editorial with the UK-based magazine **EDGE**, and is not affiliated with NeXT Software, Inc. Unsolicited manuscripts cannot be returned or acknowledged. To Roppongi, and Andy, who was in smelling distance

6

So what do they teach you at videogame school?

If you want to make games for a living, then you should listen to Claude Comair, president of the Digipen school of videogame programming. In this exclusive interview with **NG**, he gives his advise to those who'd like to turn a hobby into a profession

40

Sony's new PlayStation: If you can build a better game...

Sony's latest machine costs around $800, is matte black, plugs into a PC or Mac, and is scheduled for a U.S. release this March. Excited? You should be. Because Sony's *Yarouze* system gives you everything you need to program your own PlayStation games

mal
cuse

Pub: Sega
A stylish graphic adve
style of Warp's D, Torico is
all graphic adventure fans. Sublime
graphics, effective attention to detail, and
great story.
Rating: ★★★★

Ultimate Mortal Kombat 3
Pub: Williams Dev: Eurocom Dev.
More imitative than innovative, *Ultimate Mortal Kombat* is essentially a lazy alteration of *MK2*'s more dazzling play mechanics and characters.
Rating: ★★★ **(NG 19)**

Virtua Fighter 2
Pub: Sega Dev: Sega AM2
Sega's adaptation of its own arcade game is flawless. The depth of fighting coupled with the ease of learning moves barely stands above stunning graphic execution. A must-have videogame, and game of 1995.
Rating: ★★★★★ **(NG 14)**

Virtua Racing
Pub: Time Warner Dev: Time Warner
Virtua Racing delivers excellent polygonal racing with additions not present in the original, like a circuit mode, new cars, and new tracks.
Rating: ★★★ **(NG 12)**

Virtual Open Tennis
Dev: Imagineer
to Power Serve 3D Tennis
Virtual Open
computer AI

World Cup Golf
Pub: US Gold Dev: ARC D
A subtle interface, several
excellent graphics, effectiv
and easy-to-learn game m
this a solid one-course g
Rating: ★★★ **(NG 17)**

World Series
Pub: Sega Dev: Se
Sega's improvement
arcade style pitchin
major league stadi
the detail in every
Rating: ★★★★

Tengen
reverse, *Tama*
control the course in
t be maneuvered from
B. Weird and obscure.

Valley Golf
Dev: T&E Soft
like any second-rate golf
comes with
ntains, and

56

Every Nintendo 64, Saturn, and PlayStation game reviewed

Yes, you read it right. Over 300 next generation videogames played, reviewed, and rated in the most comprehensive 32-bit and 64-bit gaming guide ever assembled. Don't buy a game before looking here — and don't end up buying a turkey this Christmas

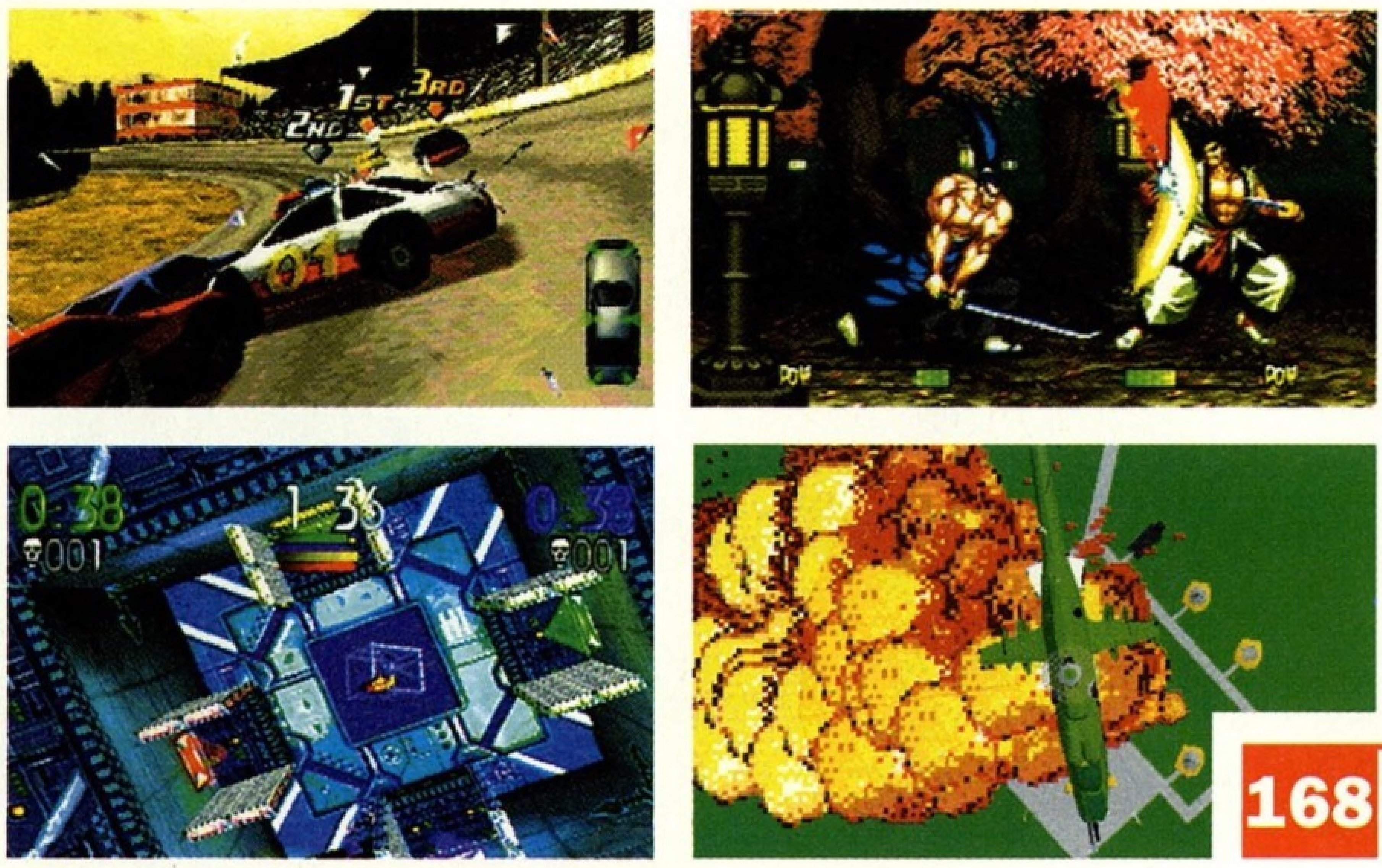
introducing

News

A field report from the PlayStation Expo in Tokyo • Highlights from the JAMMA arcade show • Europe's biggest videogame show — with no sign of Nintendo? • Joyriding: Online gaming

Alphas: 19 new games revealed

Thirty three pages of the hottest games in the works, including: *Super Mario Kart 64*, *Christmas Nights*, *PaRappa the Rapper*, *Apocalypse*, and an interview with Jordan "*Prince of Persia*" Mechner



Finals: 31 new games rated and reviewed

Do your holiday gift buying with confidence, knowing that **NG**'s crack staff of gamers knows what's good and what's bad. So read on and keep your presents from being used as coasters

6 talking

So what do they teach you at videogame school?

Claude Comair, the president of the Digipen school for videogame programmers, offers his advice to those who want to turn a hobby into a profession

14 breaking

News

All the latest information from the world of gaming

24 **Arcadia** (coin-op news and updates)
26 **Movers & Shakers** (business news)
28 **Joyriding** (online gaming news)

32 subscribing

Save cash and time

It makes sense to subscribe to **Next Generation**

www.next-generation.com

Next Generation Online

Get daily news updates, sneak previews, demos, downloadable software, and reader interaction

40 ng hardware

If you can build a better game...

Sony's Yarouze system lets anyone make a game. We show you how with tips from the pros

56 ng special

Every Nintendo 64, Saturn, and PlayStation game reviewed

Over 300 next generation games reviewed and rated

77 ng software

Alphas

Begged, borrowed, and stolen, nineteen games from around the globe: *Super Mario Kart 64* (N64); *Christmas Nights*, *Shining the Holy Ark* (Saturn); *Apocalypse*, *Sangoku Musou* (PlayStation); *D no Shokutaku 2* (M2); *Carmageddon*, *The Last Express*, *Galactic Conquest* (PC CD-ROM)

164 now hiring

When you're done with *Yarouze*, why not try playing in the big leagues? You can start right here

168 rating

Finals

We play the games so you don't have to: *WaveRace 64* (N64); *Andretti Racing*, *NHL Face Off '97*, *Destruction Derby 2*, *Suikoden*, *Pandemonium* (PlayStation); *Tomb Raider*, *Worldwide Soccer '97*, *Mr. Bones* (Saturn); *Azrael's Tear*, *Hind*, *Red Alert*, *Star Control 3* (PC); *Donkey Kong Country 3* (SNES)

192 corresponding

Letters

Our biggest ever letters section ever. Why give our readers so much space? Because we truly love you all

198 ending

Next month...

Next Generation 26 arrives on newsstands on January 14. Find out how to subscribe on page 32

Digipen is the world's first videogame school, and typically, there are hundreds of applicants for each place in its classes. So, for gamers contemplating buying a Sony Yarouze system and making games at home, **NG** talks to Digipen's Claude Comair to ask...

Photography: MARK KOEHLER

So what do they teach you at videogame school?

Sony's *Yarouze* system will enable many people have a go at programming their own games for the first time. It's an exciting prospect — but also a tough one. So what pitfalls lay in wait for each budding game programmer? What are the common mistakes that first-time programmers make? How can these problems be avoided? What should you do if you stumble upon the best game idea since *Tetris*? And what should you do if you want to turn game programming into a profession?

Claude Comair should know the answers to all these questions. As head of Digipen, the videogame programming school located in Vancouver, it's his job to know. So, if class is all sitting comfortably, over to you Claude...

The first videogame school?

NG: How did Digipen get started?

Claude: Digipen began in the late 1980s, but it was in 1991 — after we realized that we wanted to open a school for animation and videogame programming — that we approached Nintendo for some sponsorship, and they were extremely happy to lend a hand setting up the school we have today. Now we have two sides to the school, a videogame programming side and an animation side.

NG: Where have Digipen graduates ended up?

Claude: On the videogame side, we have some working at Konami and some working at Iguana, for example. These ex-students of our programming division are actually leading the coding of projects.

NG: So does the videogame side of the Digipen school concentrate on just programming? Or do students also learn game design, user interface, and the other elements that go into a game?

Claude: On the videogame side, the main focus is on programming and is mostly science-based. The other side of the school studies animation, and is therefore more focused on art. But there is a good symbiotic relationship between the students from the two camps — they actually intermix and cooperate on common projects. So often students' games will include animation coming from the animation students.

NG: So the videogame programmers don't have to learn to do everything themselves?

Claude: Well, despite the availability of students from the other department, and despite the game programmer himself or herself knowing the basics of teamwork, students still have to come up with their own ideas, their own documentation, and their own game bibles.

Sure, if they need help in specific areas, they can ask for help. But they are taught not just the technical techniques of producing games, but also story development and so on.

NG: So each student learns a little bit about all the aspects of developing a videogame?

Claude: In a team, the programmer needs to know about artists' capabilities — and the best way to do that is to attempt to do something on their own. They are programmers, but they need to be able to talk to the artist, to deal with an artist, to explain what can be done or what cannot be done on the hardware that they are using.

Better yet, they sit down and take the electronic pen, or graphic program, or the 3D package and try to attempt to do something — nothing pretentious, but just enough to know what steps the person goes through. Simultaneously, an artist should be taught about programming and what the hardware can do. Ideally, a team comprises people who have all attempted to do the jobs of others, in order to understand the position a colleague is in when asked this or that question. This way, you will achieve a more symbiotic relationship.

> **"'Why will people want to play your game?' Students have to research this question, and come up with a decent answer"**

Basically, the ideal is a creative person who is also an unbelievable scientist, but unfortunately it's not always the case.

NG: But the emphasis is on programming?

Claude: Let's say that 90% of the time our students spend learning general mathematics and computer graphics mathematics, which is more toward matrixes, matrix algebra, vector geometry, and so forth.

NG: Why such an emphasis on mathematics?

Claude: Because math is at the core of programming and understanding computers.

Game ingredients

NG: So how do Digipen's students learn about level design, gameplay, user interface, and all the other factors that determine whether a game is good or bad?

Claude: At Digipen, specialized people will teach you this. We have a very good teacher here who guides our students in this respect and reviews their proposals. All students have to present a proposal document, which is a couple hundred pages thick, trying to convince us why the game they are proposing is good. It also contains an outline of the game's components, how much money they would need to complete it, how many people it would require, what kind of machinery it will take, what kind of specialized talent, and so on.

And yes, one component we study is the interface design. We always try to put ourselves into the shoes of the player and see what psychologically or logically is most instinctive. Would you put your finger to the left first or to the right first? How would you hold the joypad, and what are the ergonomics? How would these physical ergonomics translate into an interface on the screen — would you look first to the left or to the bottom? When you click on something to quit, you should be asked to confirm that you want to quit. If you say yes, you quit; if you say no, you get music welcoming you back into the game, rewarding you.

All these things need to be thought of.

NG: It's hard to imagine that someone at home with a *Yarouze* system would think of all this.

Claude: I believe that many people doing it on their own would miss these little things — and that is why I would recommend that if you would like to pursue programming as a career, then you should seek universities and schools to guide you properly.

How to make games at home

NG: But given that not everyone can attend Digipen, what advice can you give budding programmers who want to start learning with Sony's *Yarouze* system?

What should they do first?

Claude: I would suggest that people should try to make a puzzle game, in order to test their ability in coding to begin with. Also, puzzle games show a lot of a person's creativity. If I gave you $20 million, lots of explosives, and some stunt people and said, "Go make a film," you could create a big splash on the screen that would cater to many people. But if I gave you a couple of dollars and an instamatic camera — then you would have to compensate for a lack of good equipment with a lot of imagination.

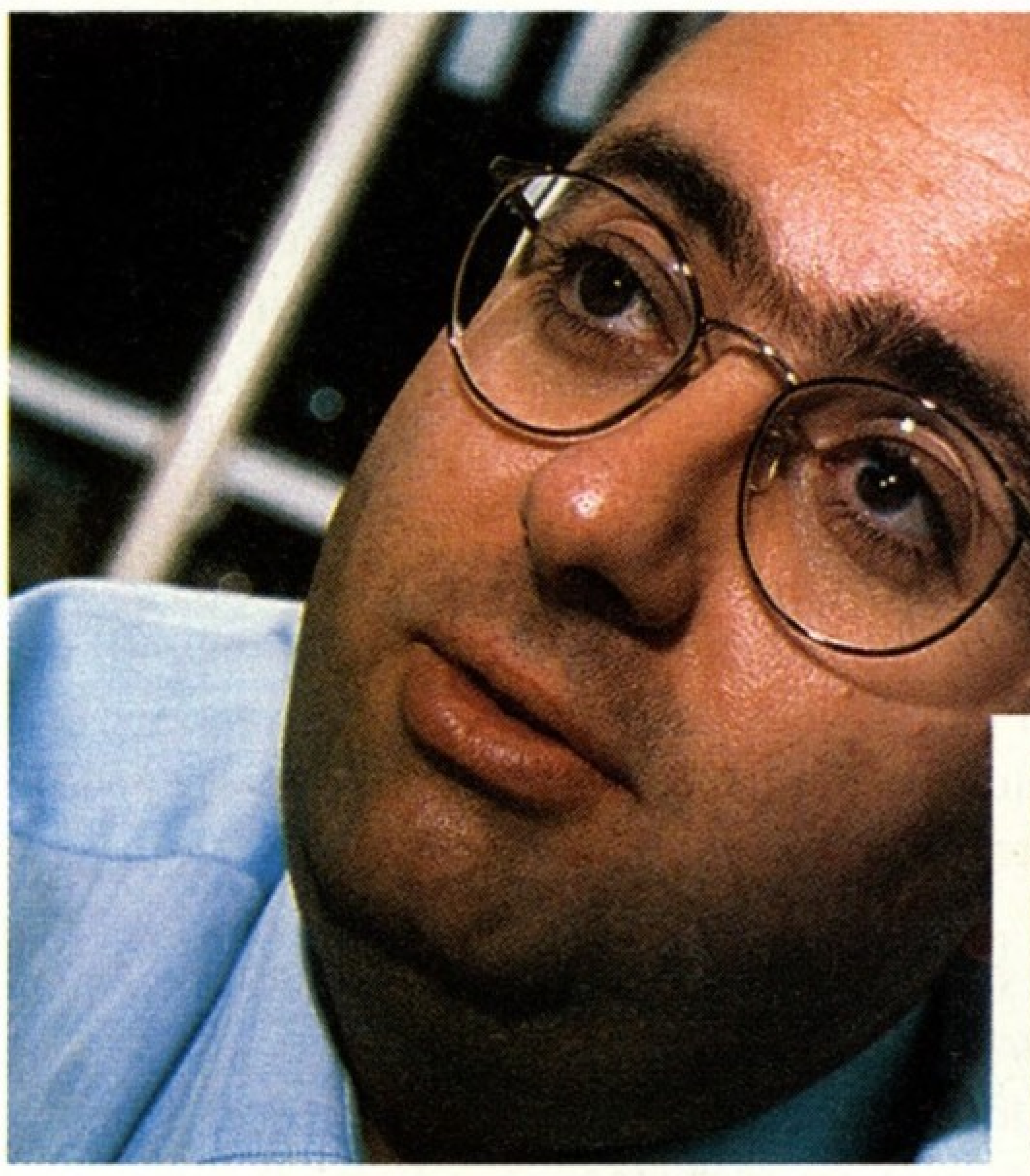
NG: And the same is true of videogames?

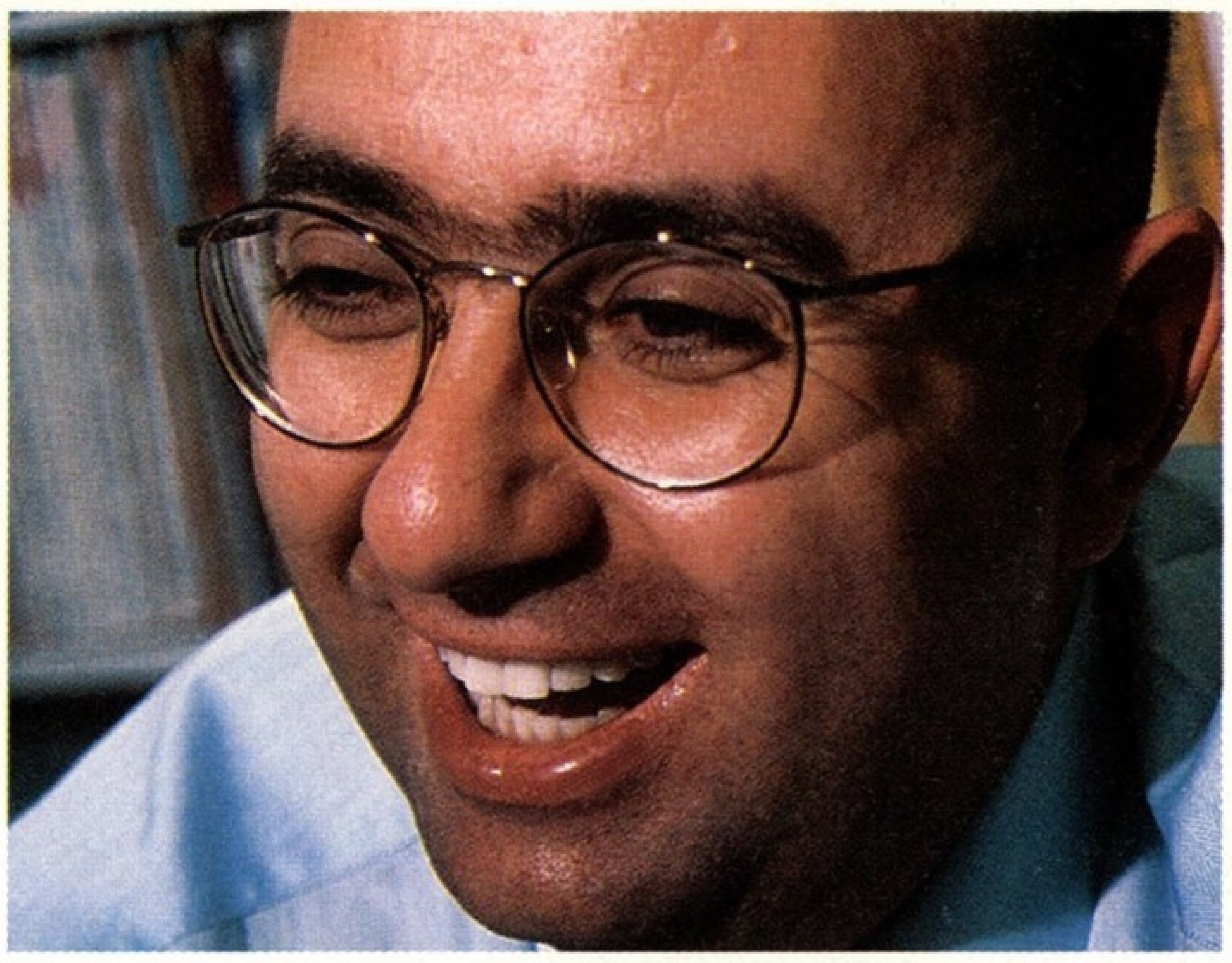
Claude: Millions of people still play *Tetris* — and yet it is a good example of a program task that can be achieved by just one person.

NG: OK, so after completing a puzzle game, what should *Yarouze* users keep in mind when they try to tackle something a little bigger?

Claude: They have to keep in perspective two things: One, that the game is made of two components living in close relationship — one cannot live without the other; Two, and a result of this, they need to study their computer sciences.

The two components that make up a game are the story and the artificial intelligence that actually drives the story and results in graphics on the screen. These two major components — art and computer code — have to come together in a very coherent way. A game without the story is not a good game. A good story without clever code turning it into a game is no good either. Thus, no matter how great a game programmer's imagination may be, the basics cannot be ignored. Game makers have to know their mathematics very well and computer science very well.





NG: *Yarouze* comes complete with many programming libraries, so doesn't this to some extent reduce the need for math and programming skill?

Claude: The danger is that games made by cutting and pasting libraries in this way may end up looking the same — and I don't think that this is what Sony intends. Certainly, it's not the best way to learn how to program games for a living.

Real programmers should be able to reproduce for themselves all of these routines and libraries of code that come with any system — and outperform them many times. Certainly it is this way if they want to become professionals. Professionals want "fat-free" code — code that is very optimized, and put together in the fastest, most economical way possible.

This sort of code is often the only real difference between a bad game and a great one.

NG: So you're saying that using these libraries is like, say, a supposed "master chef" simply heating up oven-ready meals — he's not really *making* anything. And your recommendation to someone who wants to turn programming into a profession is to learn how make their own libraries — and this means studying computer science and math?

Claude: Yes, because this way, you will fully understand the programming tools you purchase and use. You will have a full awareness of what a tool can do, and sometimes maybe you can actually bypass some of its functions and do something great. Pens you can buy anywhere, but not all of us can write poetry. Do you see what I mean?

NG: Presumably, you also need to know your computer languages well...

Claude: Obviously, you need to know your computer languages perfectly because that is the only way in which you can talk to the hardware — and this is at the heart of what you are doing. If you are talking inadequately, you immediately have a handicap.

NG: And study of computer science will help you understand how these languages work?

Claude: You need to know all your devices — all your IO boards, all you interrupts, all of what makes up the computer you're working with — in order to be able to drive the machine at the lowest level possible. You don't want to pile up too many translators between yourself and the hardware because they will slow you down.

"Pens you can buy anywhere, but not all of us can write poetry..."

NG: What other disciplines should potential games programmers study?

Claude: Physics is also important.

NG: Why *physics*?

Claude: Because most of the time games are an interactive simulation of the real world. The real world is governed by the laws of physics — gravity, motion dynamics, wave optics, and so on. And these have to be understood by the person who is trying to create that simulation.

NG: So once a budding game programmer has a great idea, has studied all he needs to study, and is about to embark on making a game, what common mistakes is he likely to make?

"If you wish to simulate the real world, then you need to understand how the real world works — and that means physics"

Claude: I believe that the biggest mistake would be not approaching a professional game company! Many game programmers forget that the best productions are usually the result of teamwork, and that a established devlopment company can offer tremendous support.

I have this problem with my students. We beg them to come to school, to do all their work at school, and to team up with other students. We beg them. And yet we have all these students who are rich enough to get their own computers — and they do their homework at home. They disappear during the evenings, they don't share information with the others, and they think they are doing the right thing. But when they come and face the results of the group, they realize how weak their project is.

The secret is teamwork...

NG: So you advise *Yarouze* users to team up?

Claude: I think they should organize themselves into a team with various capabilities and various talents and act as if they were a company. Put things into perspective — their charts and goals. And do a study of who would buy the game they plan to make and why would they buy it. The next thing you know, they are doing what every game producer does.

NG: So you don't see any new solo videogame "stars" emerging from the *Yarouze* project?

Claude: Well, it will make us more aware of who is out there — it "gives the mike" to some new people — but the 1980s era when one person could make a hit game by himself is long gone. These days games cost in the millions of dollars to produce — and no investor is going to give this money to one guy working from his bedroom. In 1996, a game will probably employ 10 to 15 people, working for one or two years.

NG: So you think *Yarouze* is a bad idea?

Claude: No, of course not. And a lot of amateur talent will be discovered this way.

But we must not give the impression that *Yarouze*, or any solo solution is like a pill that can be bought off the shelf for solving a headache or something. Game development requires lots of people in this day and age, and at Digipen we emphasize that this is not a solo adventure. It is no longer an affair where in a basement where you can solve all the problems on your own.

Selling the big idea

NG: But surely someone could come up with a good idea on their own in their bedroom?

Claude: Of course, you could come up with a good idea — but you'd still need the support of a major corporation to back you up in order for you to produce it.

NG: So if someone does come up with a great idea, should they present it to an established software publisher?

Claude: Absolutely. Unless this person is heir to a humongous fortune...

NG: And how should the game publisher be approached?

Claude: If someone is in possession of a good idea, I would recommend that they keep it very confidential — publishers don't like games everyone knows about. The publisher also needs to know that the idea is genuine and does not contain material that is subject to lawsuits.

And if the publisher says, "No, it is not something that I would like to produce," then they should actually listen carefully and not get personal or angry. But ask themselves, "Why does this person not want to put in the money?" People might realize that they misjudged the value of their side of the story. If you are a person that is constantly misunderstood by the population of publishers, then maybe that should be a message to you that maybe you're not answering their need.

If the publisher says "Yes," however, then you're in luck. ng

For more Digipen information

If you would like further information about the Digipen school, write:

5th Floor, 530 Hornby Street
Vancouver
V6C 2E7
BC
Canada

Sony plans to reintroduce the pioneering spirit of game development's "good old days" with a $750 PlayStation development system that lets *you* make your own games. A **Next Generation** exclusive

Sometime in the first quarter of 1997, Sony Computer Entertainment America (SCEA) will introduce a new product into the U.S. market called *Yarouze*. Available only from Sony, it will cost around $750 (final pricing has yet to be set) and contain a special black PlayStation, a serial cable for connecting the system to a PC or Mac, and a CD filled with PlayStation programming libraries, a C compiler and debugger, and other development tools. For the first time ever, consumers will be able to program their own games for a console system. If you don't realize what an important development this is, you probably shouldn't be reading this magazine.

Over the next 10 pages, **Next Generation** will present the first English language, in-depth report on the system. What it is, how the *Yarouze* program will work, and what it means for gamers and the industry as a whole. As a service to our readers (many of whom will no doubt order a system as soon as possible) we have also included tips on game design issues from the pros. Enjoy...

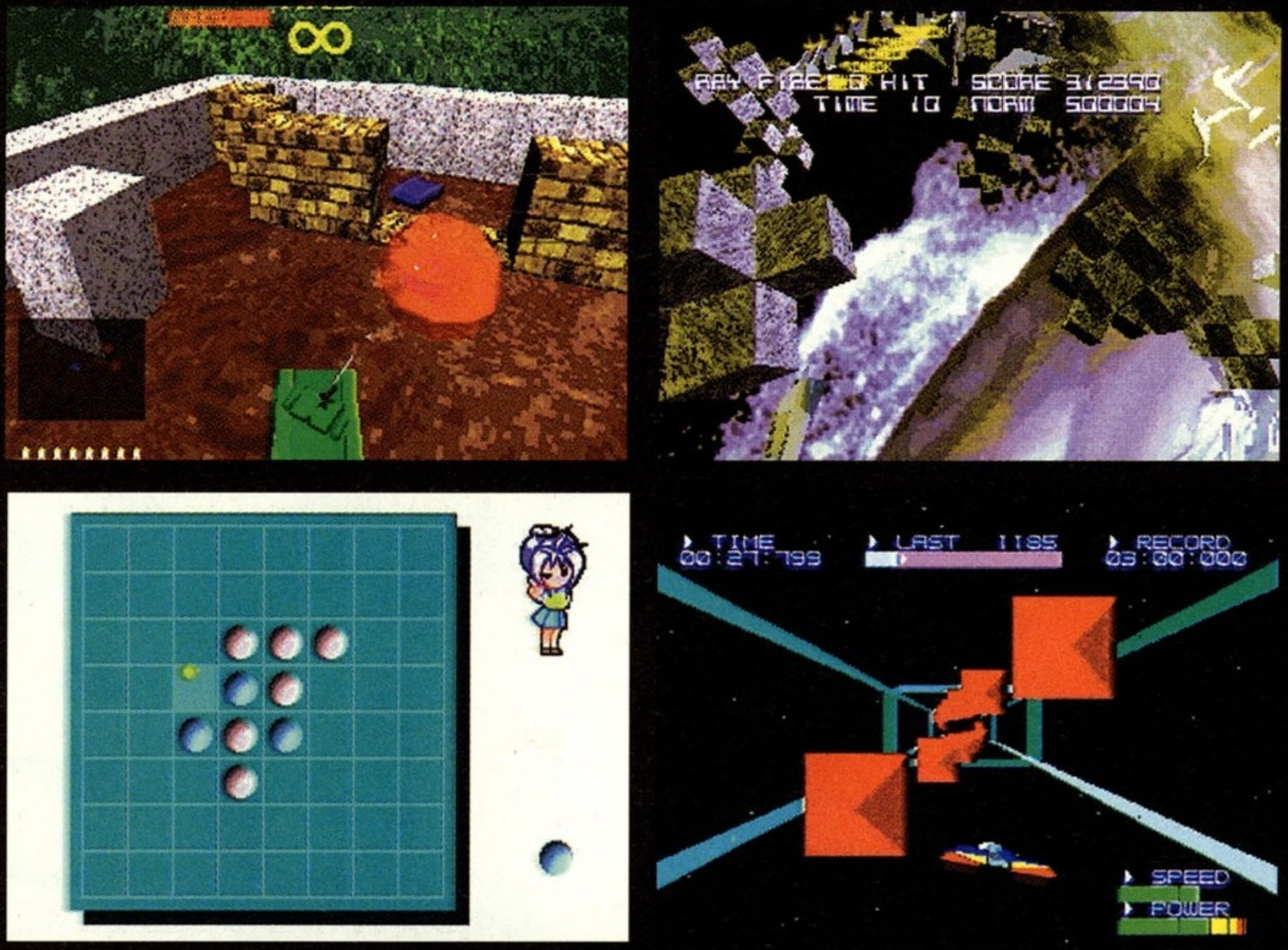
If you can build a better game...

What's the **big deal** anyway?

Sony wants to return some of 1997's game development back to the way it was 15 years ago, an age when gamers made games themselves, for themselves. It's a decision that could save the industry — and turn some kids into programming stars

Yarouze — a Japanese word that roughly translates to "let's do it!" is one of the most significant developments in the game industry in years. This innovative program, conceived by PlayStation creator Ken Kutaragi, will, for the first time, allow enthusiasts to create their own console games.

Released in Japan in late 1996, it will ship in the United States sometime in the first quarter of 1997 (a release date was not set at press time). Basically, *Yarouze* is a program that includes software for PC (or Mac), hardware in the form of a specially modified, black PlayStation, and membership of a development club that takes the form of a private Internet website. With *Yarouze,* hobbyists can create software for PlayStation for under a thousand dollars. No official price point has been set in the U.S., but **Next Generation** expects the system to sell for about $750.



These efforts from the *Yarouze* program in Japan are very impressive for first-generation titles. Sure, they're mostly 3D shooters (always a shareware favorite), with the odd puzzler thrown in for good measure, but compared to most PC shareware efforts, they look great

The CD that comes with *Yarouze* contains a C compiler, a debugger, tools for converting graphic and sound files to PlayStation format, and the PlayStation development C libraries — the same libraries that make developing for PlayStation so much easier than for Saturn, Nintendo 64, or even PC.

The system is not quite as robust as the professional level PlayStation development system. Any software created for *Yarouze* must fit entirely in the PlayStation's 3.5MB of RAM (2MB main RAM, 1MB VRAM, and 512K sound RAM), so libraries for things like streaming video or data off a CD are not included. How much of a handicap is that? Not too much — *Ridge Racer* fits entirely within system RAM and could be done with *Yarouze.*

The website will provide a place for *Yarouze* owners — members is a better word — to ask questions of Sony technical support staff, upload and download software, and talk to each other about development issues. The web site currently serving the Japanese *Yarouze* program also sponsors members' homepages, but it is not known if the U.S. site will at this time.

$5 billion of game software and hardware will be sold in the U.S. this year. It's an industry dominated by corporations. But it wasn't always this way...

Some hardcore gamers may pick up *Yarouze* just to play other people's games

To fully understand the potential significance of *Yarouze,* some history is required. The video and computer game industry is huge — the latest figures indicate that over $5 billion of hardware and software will be sold in the U.S. this year. Increasingly, though, it's an industry dominated by a few massive corporations. But things weren't always this way.

Fifteen or twenty years ago, sure, one's console options were dictated solely by one or two companies, but with personal computers like the Apple II and Commodore 64, things were much different.

For PC games, no prepackaged software industry existed to speak of. The best you got was a floppy disk and some mimeographed instructions, tastefully packaged in a Ziplock bag. The person ziplocking the software was likely the same person who designed and wrote the program *and* did the art and music.

There was an excitement present during that time, a feeling that one was participating in something new and different. The truth is less romantic — a lot of that early software, frankly, sucked (there isn't a very large quality control department in a one-person shop). Still, it was rarely predictable.

With no established genres or gaming paradigms, your $10 Ziplock-encased game was as likely to contain a shooting game as it was a text-based game where you controlled truffle sniffing pigs. And if you didn't like a game, it usually wasn't too much of a stretch to either go ahead and modify the code of the game you just bought, or rewrite the same game in a style more to your liking.

As the industry grew and matured, computers became popular with people more interested in simply using them than programming them. Companies that started in bedrooms as hobbies, producing software for other hobbyists, grew and began producing software for non-hobbyists.

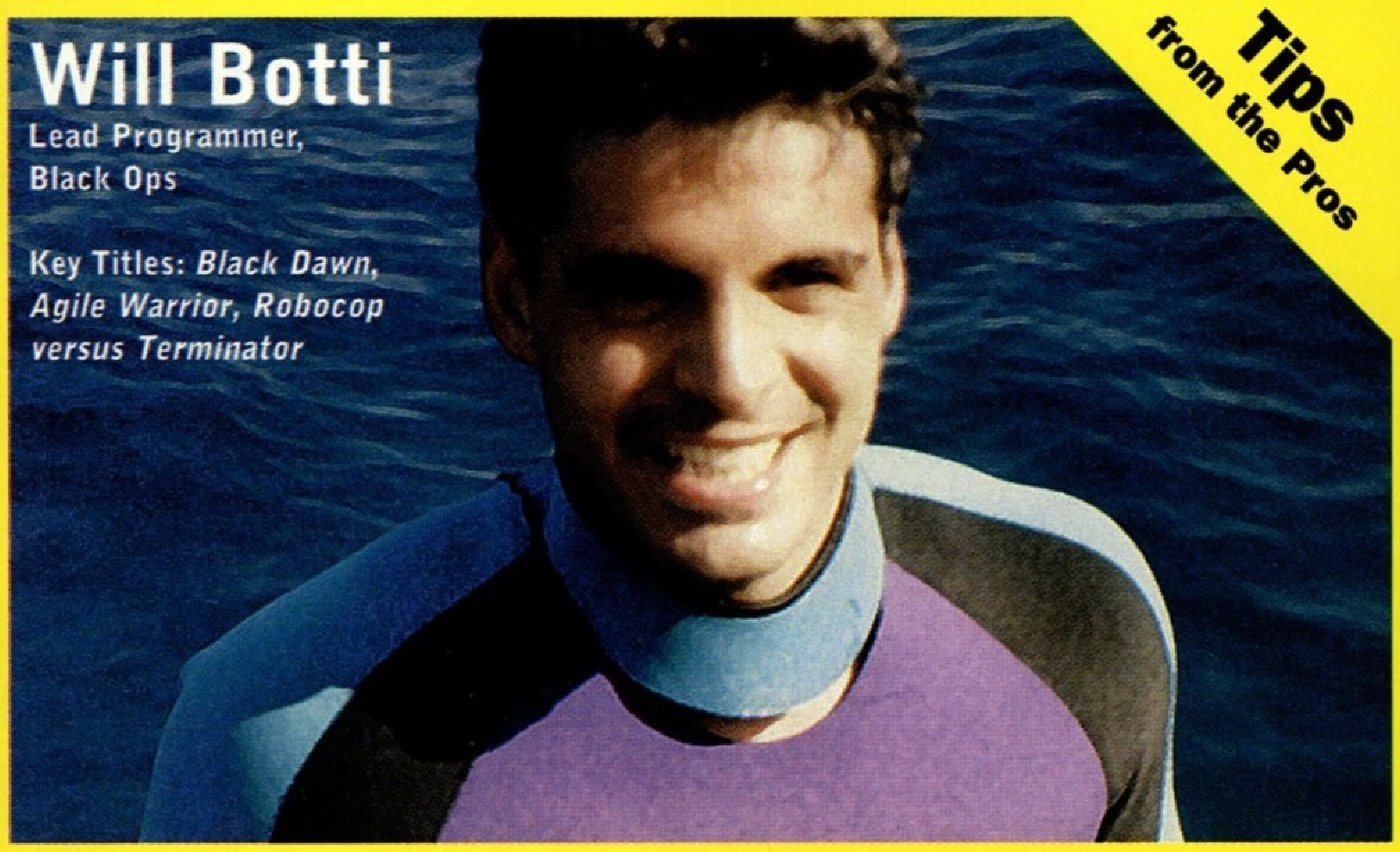
That was a good thing, because they started quality assurance departments, conducted focus group tests, and began to give gamers exactly what they wanted. There were some downsides to this (largely inevitable) process. As games grew in size and complexity, releasing a game became a larger financial risk (a game that takes a year to create costs a hell of a lot more to develop than a hack knocked together in a weekend). To ensure a good return on investment, most games released were in one of a few specialized genres, and innovation began to slow. Still, many hit games, like Peter Moleneuyeux's *Populous*, were still the product of one person or very small teams.

Then came *Myst*, and with it a flood of multimedia-equipped PCs. Despite *Myst's* merits (or lack thereof), it radically changed consumers' expectations of what a computer game should be, graphically. Point blank, expectations shot through the roof. Although a dedicated amateur might have a shareware hit, producing anything approaching commercial quality suddenly required far more dollars and time than most hobbyists had. This has only exacerbated the problem of decreased innovation. Id's *Doom*, released just before the "CD-ROM revolution," was perhaps the last title that was successfully produced "in the garage."

"Since CD-ROMs came in, the audience has demanded so much that you can't really innovate," says Jordan Mechner, creator of the classic *Karateka* and *Prince of Persia* titles. "To have a bright idea and follow it through to the point where it is a sellable game, you now need millions of dollars and a team of trained professionals." Understandably, companies are reluctant to spend millions of dollars on anything but sure things, in well-known genres. "The game industry has been paralyzed — it's really hard to be creative," laments Mechner.

Say what you want about how the old classics aren't what they're remembered to be, at least their designers were not afraid to try new things. Today, with the majority of video and computer games vigorously locked into a few genres, innovation in gameplay has largely been replaced by mere technological advances. To be sure, sometimes advances in technology can advance gameplay (*WaveRace*, and *Tomb Raider* would not have been possible two years ago), but *Quake* is not much more than what *Doom* would have been if P166s had existed in '93.

If the game industry is to avoid sliding into the same paradigm as TV (every year it's the same old rubbish, just with different actors), with any innovation or experimentation crushed by market pressures, it needs some kind of



Tips from the Pros

Will Botti

Lead Programmer, Black Ops

Key Titles: *Black Dawn, Agile Warrior, Robocop versus Terminator*

Will Botti has been programming games since he was in the fifth grade

Ten things a 3D shooter should include:

1) Easy controls that perform hard-to-do maneuvers
In *Black Dawn*, controlling your angle of attack is easy — just up or down on the D-pad. In a real chopper, you'd need to worry about collective, blade pitch, wind, and so on. Easier is generally better.

> **"If you can't make a game with the tools that Sony provides, you're an idiot and don't belong in the business"**

2) Ability to destroy entire environments
In *Agile Warrior*, many gamers really enjoyed being able to deform the environment and create canyons through mountains. Not realistic, but who gives a shit?

3) Secret stuff
People love secrets, be they hidden areas or hidden powerups. Secrets help give the joy of discovery, and the power of that knowledge. Try playing a *Quake* deathmatch with someone who knows all the secret areas — you're gonna get waxed.

4) Large, dynamic 3D explosions
3D games should have 3D explosions.

5) Large, high-detail onscreen objects
In 2D fighting games, it was the "big sprite" that ruled the day. *Zangief* filled our screens and we loved it. Just because games are in 3D and a lot of the action may be 'into' the screen doesn't mean things have to be small. Make 'em big.

6) Vectored bullets
Bullets fire from the enemies' position and "vector" or move towards the player. Sounds simple, but lots of games miss the "avoidathon" gameplay that vector bullets provide.

7) Detailed collision between objects
Compare two game engines — *Tekken* vs. *Toshinden*. The collision in *Tekken* is superior, so the *Tekken* player is more empowered. Result? Read the sales reports and see!

8) Damage by degrees
The more levels of damage an object has, the more realistic/immersive it is. Good examples are the awesome submersible-cracking in *Earthworm Jim*, or the player ship in *Warhawk*. When a player gets hit in a 3D game, you should: show a "ping" animation (bullet sparks), deform the geometry of the model, change the texture map to destroyed state, or reduce the controllability of the player vehicle (carefully).

Remember, though, there's a fine line between fun and frustration.

9) Payoff for advancement
Too many games end with a static screen. Give the player a real reward. The best example is in *Tekken 2*.

10) Two-player option
If possible, make the game two-player. I'll take playing against someone in the same room over playing the computer or even someone on the Internet any day! I want to see my opponent react and get pissed, and then try to get even!

Once you've pulled the system out of the box and removed the packaging, this is what your $750 (or so) buys you. The really cool bits — the PSX C libraries —are on the silver CD

jump start, a way of bringing new designers into the mainstream, without being marginalized into the shareware ghetto or stuck being junior programmers on the latest *Doom* clone. *Yarouze* may very well be that jump start.

“This is one of the best things that's happened in a very long time,” says programming evangelist and *Earthworm Jim* creator David Perry of Shiny Entertainment. “Creating a videogame requires passion, and passion is something that a lot of people have — but a lot of people with passion can't get anywhere near a development system and get through the rules and regs of making a game.”

Industry veteran Eugene Jarvis, creator of *Defender*, *Robotron*, and *Cruisin' USA*, is equally enthusiastic. “It'll be wild — imagine linking up all these lone wolf designers over the net to create a huge meta project — if you can get any lone wolves to agree on anything. This is an awesome development.”

No doubt. By enabling enthusiasts to create their own games, Sony has vastly lowered the barriers to entry for would-be developers. Also, it has created an environment in which new ideas, unrestrained by market pressures, can be tried and tested, the best of which may advance the state of the art of the industry.

“Reducing the cost of entry into the industry will bring people into it,” said Mark Cerny, president of development at Universal. “The whole reason *Disruptor* exists is because an enthusiastic hobbyist could start development on 3DO, since it had cheap development hardware. We saw the prototype running on 3DO and agreed to fund development on PlayStation.”

Yarouze may not change the gaming world. But someone who gets their start on _Yarouze_ might

He continued, “What we'll see primarily is hobbyists creating purely passion-driven games with no consideration for commercial success. I think we'll see some great games we wouldn't have seen otherwise.”

Despite enthusiasm from the developer community, skeptics may ask some questions about the true importance of *Yarouze*. Doesn't the possibility for independent, enthusiast development already exist on the PC? Other than *Doom*, what has the shareware world brought us, except for a myriad of lame shooters? Why should it be any different on PlayStation. In short, what's the big deal?

The answer to this question reveals, in large measure, just what is so exciting about the initiative as a whole, especially considering that most *Yarouze* members will have already tried their hands at game coding on the PC or Mac.

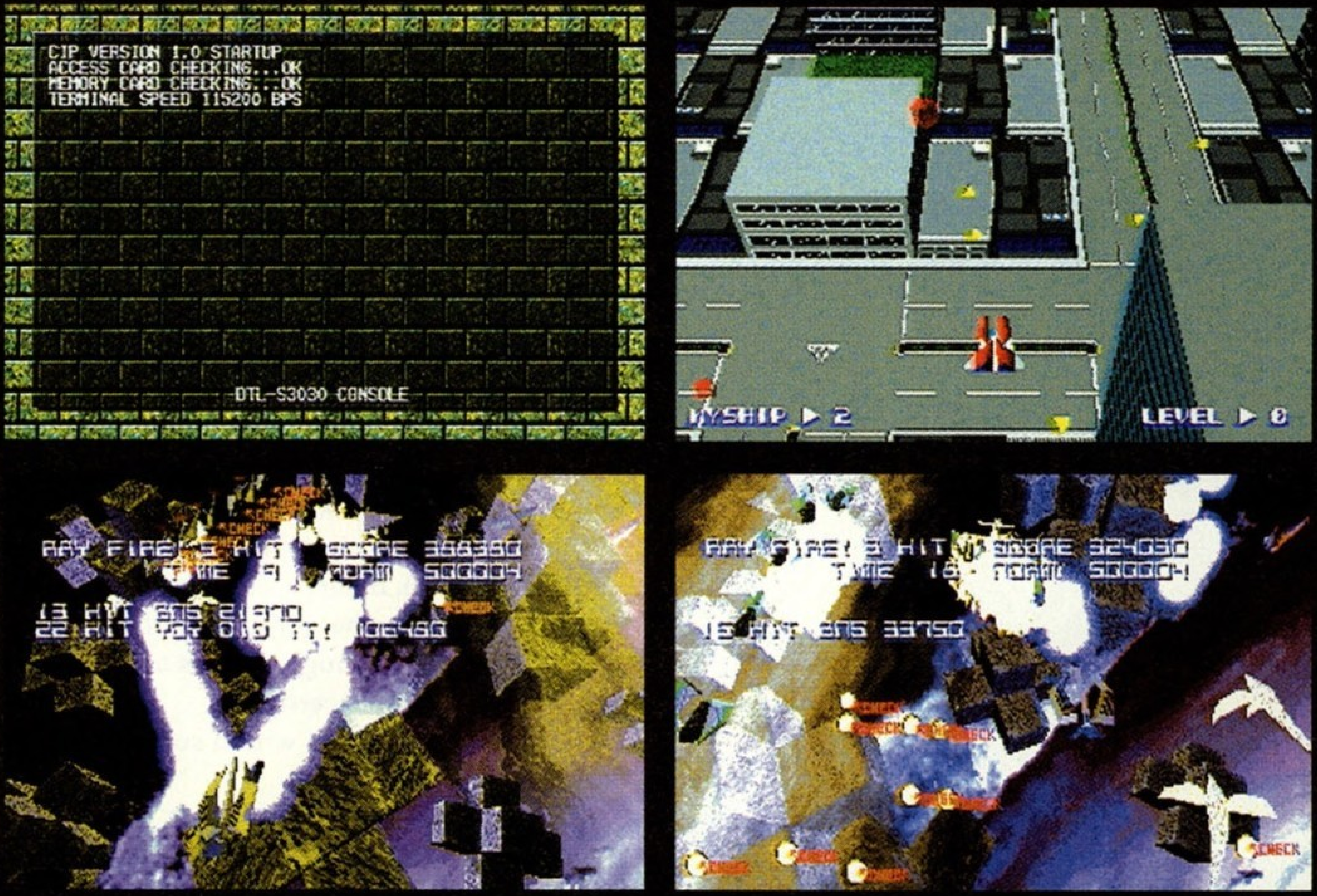
First, this is the first time that anyone will be able to develop for a console system without a significant cash outlay, which will give enthusiasts easy access to a single standard hardware platform. Instead of attempting to develop for a lowest common denominator of hardware (or worse, a Windows API), people will be able to concentrate on exploiting the PlayStation hardware as fully as possible.

The program libraries will enable people to get up to speed in complex 3D development far faster than they would be able to on a PC. The libraries will also aid more novice developers whose own 3D skills might be sub-par at first.

Also significant is the support level from Sony and fellow *Yarouze* members that will be available on the website. For the first time, hobbyist game developers will have someplace official to go to get information on issues. It is also expected that the website will, as it has done in Japan, become a repository for source code that will enable faster learning of just how to program the system. The importance of support from Sony, particularly for novices, cannot be overstated.

According to Will Botti of Black Ops, “In a lot of ways, PlayStation's success is directly related to the caliber of Sony's ‘teachers’ — the staff at SCEI and SCEA — and their ‘teaching materials’ — PlayStation development kits and artist/sound tools.” With Sony extending this level of support to include novices, software produced for *Yarouze* may well outstrip the quality of comparable PC software.

Even the 3.5MB limit will probably not be too much of a factor — remember, before the advent of CD-based systems, 3.5MB was considered quite a lot of storage space. “By



```
CIP VERSION 1.0 STARTUP
ACCESS CARD CHECKING...OK
MEMORY CARD CHECKING...OK
TERMINAL SPEED 115200 BPS

DTL-S3030 CONSOLE
```

A shot from the _Yarouze_ startup screen (top left), and more shots of nearly complete games. The Japanese _Yarouze_ site is also home to many PS-X demos and hacks

limiting it to 2MB [main system RAM]," says Eugene Jarvis, "the focus needs to be on gameplay — not graphics."

Most importantly, though, the PlayStation provides a new horizon for enthusiast level programmers to explore and hack around in — a new world to conquer, exploit, and make their own, which is where most of the fun of programming, particularly at the hobbyist level, comes from.

Of course, the biggest benefit of the crossover between PC and *Yarouze* enthusiast developers will not be for them, but rather from them. For the first time, we will see innovative, hobbyist level development on the consoles. Game ideas that would never fly as commercial titles, but which may work far better on console than on PC, can now be implemented and enjoyed by other *Yarouze* owners.

Another exciting element about the system is that, unlike commercial PlayStation software, no approval process exists — anything goes. If you want to do *Lemonade Stand* on the system — go for it. "This is totally against the conventional wisdom that product must be controlled," says Jarvis, "It's neat to give access to anyone to do anything."

Don't expect the system to sell solely to programmers, either. In Japan, a large number of artists and designers have also purchased *Yarouze*, looking to team up with programmers to create games. **Next Generation** also expects some hardcore gamers to get *Yarouze* simply to play the games that others have created — as well as Japanese and European games (the black PlayStation has no teritorrial lockout).

The *Yarouze* system opens up a totally new range of possibilities for users, and has the potential to greatly invigorate the software industry as a whole.

So what does Sony get out of it? "If *Yarouze* were to be judged on marketability alone, I would say it's going to be a a big flop," says Will Botti. "Kids want to play games, not program them. I don't see *Yarouze* as a big profit center. But if we look deeper, we can see the reasoning behind Ken Kutaragi and Terry Tokunaka's plans for the system. In the same way that the extensive developer support helped endear top game designers to the platform, I see *Yarouze* as a way to endear the younger generation of hackers/game enthusiasts to Sony's PlayStation methodology: "Code in C, use our tools, don't reinvent the wheel, focus on games, not the friggin' video drivers, and so on."

In short, *Yarouze* will let enthusiasts concentrate on creating games, instead of being bogged down in the minutiae of programming for an OS like Windows that must also be able to run printer drivers and screensavers.

Ultimately, *Yarouze* may not change the videogaming world. But it could be someone who gets their start on *Yarouze* that does. And that's why it's so important.

Tips from the Pros

Dave Perry

President,
Shiny Entertainment

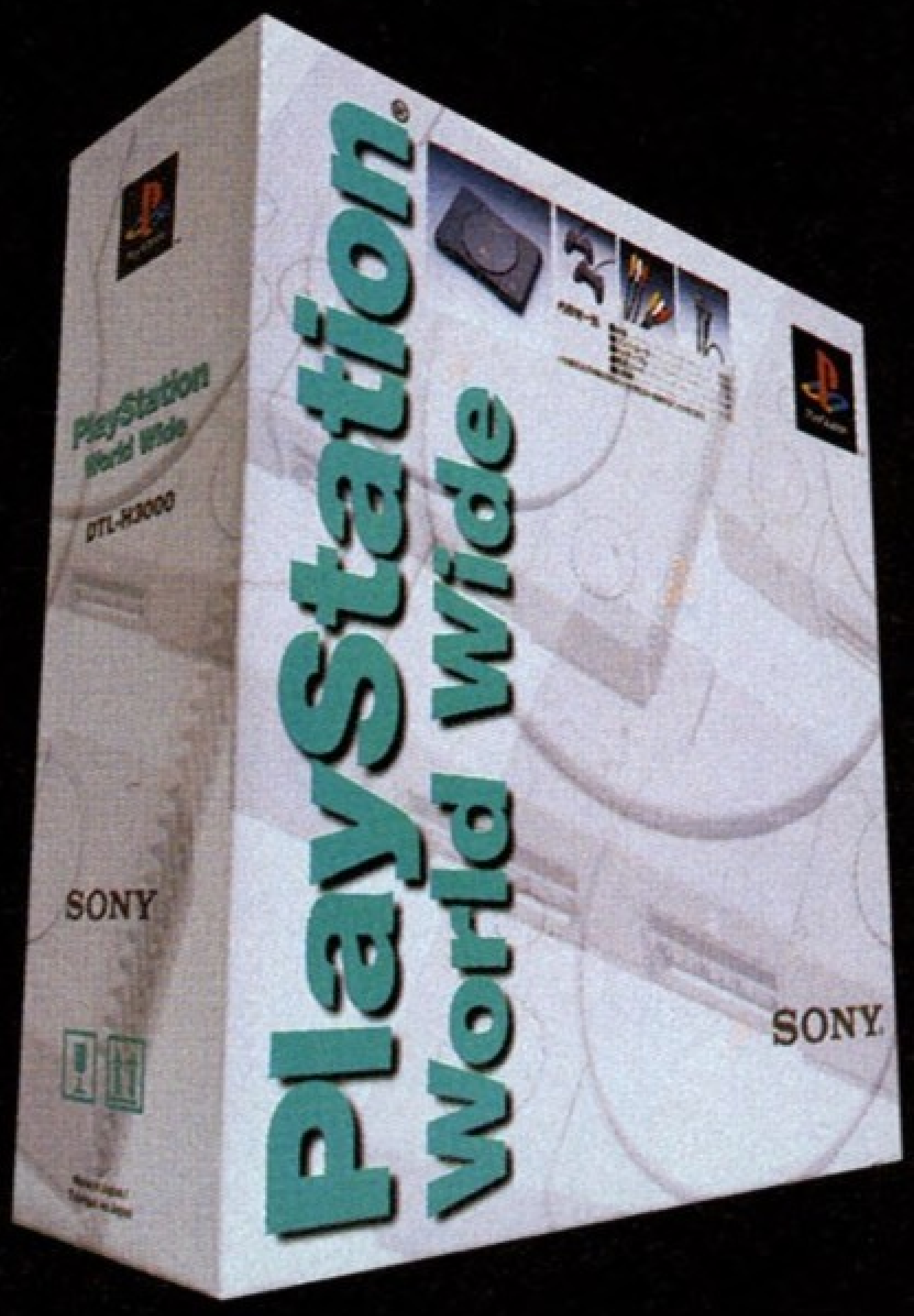
Key Titles:
Earthworm Jim, Aladdin, Cool Spot, Wild Nines, Global Gladiators, MDK

Dave Perry got his start writing game program listings for videogame magazines

Ten tips for game programming novices

> **"This is one of the best things that's happened in a very long time"**

1) Starting: Challenge yourself and don't stop till you work it out. Getting a program that you designed to work is like winning a game (the same sort of feeling). You *will* get *very* frustrated, but frustration is good, because it is such a relief to get that damn thing working.
2) Learning: Go straight to the biggest bookstore you can possibly find. A good reference book can fix problems very quickly when you are working late. (Also use the Internet.)
3) Gameplay: Don't ever take control away from the joypad/keyboard unless your really want to piss off the player.
4) Gameplay: When you press jump, make him jump. Fight animators or anyone who tries to get you to do anything else. Instant response is *key*.
5) Personal: One of my best tricks is to make every damn possible thing random. If something repeats (for example if your character looks left and right) don't make it ping-pong in perfect timing like a metronome. Always slip in randomness so that something that does repeat never looks the same twice. Nothing in your game should move to a "beat."
6) Technical: Tables are the fastest way to program just about everything. If I have a bird that is flying and it randomly decides to swoop down, should I do a bunch of math to pick a random number? Or should I just grab the next value from a previously created table?. Your code grabs a value each time it wants a random number, if the number is greater than one hundred (>100), then make the bird swoop. It's as simple as that. When your program gets to the end of the table make it go back to the start. (No gamer will notice that the random numbers are being used again! —That's the beauty of it.)
7) Psychology: The best bit of a video game is winning. When a player does well, pat him on the head. It is *much* more important than you probably think. If he obviously screwed up, make him feel it was *his* mistake. This will make him think that with another go he will win. The player should *always* blame himself, not the game for failure.
8) Competition: Speed! If you can work out a way to get the screen being updated 60 times a second and someone else can only get 4 times a second, you have totally kicked his ass. This is another enjoyable part of programming. Speed comes from being clever and *not* doing things that other programmers do because *you* know that gamers won't notice. The quickest way to a gamers heart is to give him something smooth and sexy.
9) Business: If you want a job, make a great demo, send it to every publisher and team you would like to work for. Tell them that you have sent it to every other publisher and team. This means if they like it, they will move very, very fast. Good people are *very* hard to find, so if you are great, then apply for a job through our web page: http://www.shiny.com
10) Long-term: Sure, learn C to start. But never, never, ignore assembly language, it will never, go away. Somebody somewhere will always use asasembly language to make you look slow. At Shiny, our 3D games use really advanced RISC assembly code and heavily optimized Pentium assembly language. We cannot afford to be beaten. It should be your long-term goal too, and if you master these skills there is no team in the world that won't want you.



Out of box experience

So what do you get in the *Yarouze* box? Here's the official rundown of all the bits and pieces, plus all the optional extras you'll want to make full use of the system

Yarouze comes with everything you need to make software for PlayStation.

The *Yarouze* system itself is almost identical to a standard PlayStation, with the exception of different boot ROMs, the lack of a territorial lockout, and a radically different encryption scheme. (And, of course, it's black.) It differs from a professional level development system in that it has less RAM (just the standard 2MB main memory that comes with every PlayStation, and no extra RAM for debugging), and it doesn't have a CD-ROM emulator or come on a PC board.

Yarouze will come with a PC CD-ROM as well as a key disc for the PlayStation, and some manuals. The manuals are reference guides for the libraries and some PlayStation specific information, but if you're not a programmer now, don't get your hopes up — they aren't going to teach you C. Assuming you already program, though, you'll find them a well laid out guide to getting started with *Yarouze* development. On the PC CD-ROM will be C language tools, libraries and possibly code examples, although that was not determined at press time.

The standard tools supplied include the GNU Tool Chain, a Compiler Assembler, Linker, and GDB debugger. These are all "pretty direct ports of the existing Free Software Foundation tools," says Bill Rehbock , VP of R&D at SCEA. "They are a great starting point because they exist in some form for just about every platform and microprocessor in the world and most programmers are reasonably familiar with them."

Yarouze members will also receive utilities that enable conversion of standard audio and graphic files to PlayStation's format. It is expected that several third parties will also include cut-down or trial versions of their software: Expect at least a fully working demo version of one 3D package, as well as a texture-map editing utility. Also on the CD will be a nearly full compliment of the famous PlayStation development libraries. What's missing? First, some low-level model data support, which would make it pretty difficult to do something like *Tekken* or *Toshiden* right now. This support may be added at a later date if there is enough user demand. According to Rehbock, though, "the primary functions that are missing relate to CD access. The system needs to have our *Yarouze* CD in the black PlayStation itself to work, and we don't allow user-created CDs to be read. Also, the libraries are set up to be more akin to run-time libraries instead of getting compiled in with the user's code. This reduces the transfer time to the PlayStation when the user is debugging or wants to play the game."

The serial cable (left) connects your PC to the black PlayStation, but not (surprisingly), through the standard expansion port. The access card dongle (right), thwarts pirates

Because *Yarouze* needs to be pirate-proof, a special key disc and a hardware dongle that fits in memory port 1 are required when running programs that were loaded from a PC (obviously, the key disc is not necessary to run standard software in the black PlayStation). This limits the available space for programs to 2MB main system RAM, as well as the 1 MB VRAM and 512K sound RAM. Users will also not be able to read from the PC hard drive while running their programs.

In addition to preventing piracy, this means that games cannot be very large, as well as ensuring that the *Yarouze* program will not compete with the professional development kit.

However, a limit of 2MB RAM is not as bad as it may at first seem. First, quite a lot can be done in just 2MB RAM — not only can an extremely robust game engine run, but a complete game can be executed in RAM as well. *Ridge Racer* fits entirely within this 2MB limit, as would a single level of *WarHawk* or *Twisted Metal*. In short, 2MB is more than enough space in which to make a game engine and a fairly robust game — *Yarouze* titles will not be as restricted as the 2MB limit may imply to the casual observer.

Also, and more importantly, the only reason anyone would ever say that 2MB RAM is small in the first place is because developers have been spoiled over the past few years. Not so long ago, developers were creating classic games in 16K. Even given the overhead that coding in C language (versus assembly) adds, anyone who says they simply cannont create a good game or game engine in 2MB must be a very sloppy programmer.

Granted, graphics, particularly textures, may suffer with only 2MB main RAM, but this will force hobbyists to focus on the most important part of a game, the actual gameplay. Considering the most exciting aspect of the *Yarouze* program is the potential for novel, innovative games, a lack of graphic sophistication should not be that problematic.

Probably a more signifigant barrier to developers will not be the RAM limits, but the tools that come with the system. This is bare bones C and debugging folks, and anyone who has ever used a well-integrated development environment is unlikely to be thrilled by the prospect of using some of the tools supplied for *Yarouze* by Sony.

Enter Metrowerks, creators of the award-winning *CodeWarrior* C development environment. Metrowerks (http://www.metrowerks.com) came to fame with a series of PowerPC native development environments for the Mac. In October, Metrowerks released a version of *CodeWarrior* for PlayStation that runs on Mac or PC.

CodeWarrior for PlayStation costs $499, but it features such a nice feature set and devlopment environment that any *Yarouze* member who can afford it should buy a copy. It features an integrated development environment, GUI project manager, file-format converters built directly into the project manager, and drag-and-drop editing (as well, of course, as a full NIST certified ANSI C compiler, and a C++ compiler that is in track with the emerging ANSI/ISO standard).

If you plan on doing any serious artwork at all, you'll also want Adobe's *Photoshop* and probably some 3D app, too, although the 3D demo included on the CD may be enough. A sound utility may also be desirable.

Despite the RAM limitations, *Yarouze* is easily the best console development system deal of all time, even if most users will need to tack on an additional $500 for *CodeWarrior* for PlayStation.

Since the dawn of video and computer games, people have been saying, "I could do a better job than that." Now, through an innovative program that got its start in R&D, not the marketing department, Sony has given gamers a relatively easy and low cost way to prove it.

In Japan, *Yarouze*-authored software has run the gamut from shooting games to *Othello* to particle generator demos. There have also been memory card utilities written (so people can trade game saves over the Internet). Probably the most impressive "utility" is a hardware emulator for a 16-bit console system! The point is, gamers in Japan are already creating programs with *Yarouze* that Sony never expected.

What kind of games will hobbyists and enthusiasts in the U.S. make? That's up to you, but one thing is clear: Sony has delivered its promise of bringing the power of the PlayStation to the people.

Jordan Mechner

President,
Smoking Car Productions

Key Titles: *Karateka*, *Prince of Persia*, *Last Express*

Tips from the Pros

John Eaton, Smoking Car's "business guy" added tips of his own at the end of Jordan's

Ten things to do to do to ensure a more playable game

> **"A story is a series of events — a game is a series of actions. Actions not taken by the player don't count as actions."**

1) The only significant actions are those that affect the player's ability to perform future actions. Everything else is bells and whistles.
2) Make a list of all the actions that the player can perform in the game, take a cold hard look at it and decide if it sounds like fun to you. If the list is boring, the game probably will be too.
3) In each situation ask yourself: What are the possible actions the player can take? If there are only two, it's weak. If there's only one, it's not even action. If a non-interactive sequence shows the player's character sneak into the compound, clobber a guard and put on his uniform, the player's action is "Watch non-interactive sequence." Giving the player one chance to click to clobber the guard or die isn't much better.
4) Design a clear and simple interface. The primary task of the interface is to present the player with a choice of the available actions at each moment and to provide instant feedback when the player makes a choice.
5) The player needs a goal at all times, even if it is a mistaken one. If there is nothing he wishes to accomplish, he will soon get bored, even if the game is rich with graphics and sound.
6) The more the player feels that the events of the game are being caused by his or her own actions, the better — even when it is an illusion.
7) Analyze the events of the story in terms of their effect on the player's goals. Anything that moves him closer to or further away from a goal, or gives him a new goal, is part of the game.
8) The longer the player plays without a break, the more we build up his sense of the reality of the world. Any time he dies or has to restart from a saved game, the spell is broken.

Alternative paths, recoverable errors, multiple solutions to the same problem, missed opportunities that can be made up later, are all good.

9) Don't introduce gratuitous obstacles just to create a puzzle. All plot twists should mean something for the story. If the game requires the player to drive somewhere, don't also require him to fill up his tank with gas first — unless maybe the tank was deliberately drained by an opponent...
10) As the player moves through the game, he should always have the feeling that he is passing up potentially interesting avenues of exploration. The perfect outcome is for him to win the game having done 95% of what there is to do, but left with the feeling that there must be another 50% he missed.

John Eaton's "Top Four Tips for Business"

1) Get yourself a good intellectual property lawyer who has experience in the industry.
2) Make sure you haven't given away any rights except those specified in the contract.
3) Be sure to work with a publisher that's not going to go bankrupt.
4) Take the budget, triple it, then add $1 million. Double your schedule, then add 6 months.

The company line

It's great for gamers, but what is Sony getting from *Yarouze*? **Next Generation** went to Sony Computer Entertainment America's Foster City headquarters to find out

Next Generation spoke to Sony VP of Third Party and R&D, Phil Harrison, VP of Marketing Andrew House, and VP of R&D Bill Rehbock to get Sony's perspective on *Yarouze*

NG: Explain the reasoning behind *Yarouze*.

Phil: Philosophically, videogame systems are closed systems that don't allow the consumer to play around and do things with the software. And so you could argue the last time the consumer has been able to do that was with the Commadore 64 and the Apple IIgs, which means that there is a whole generation of consumers who have an interest in getting involved in the sort of gritty side of development, but can't do it.

NG: Well, you can on the PC...

Phil: ...but the PC is costly and complicated and is a moving target, so your average consumer is going to find it very difficult to get involved with.

One of the reasons for doing this program was to bring back the good old days of home development — the idea of hobby development, of two friends in the garage or in the spare room coming up with the next *Boulderdash*.

Obviously there are huge benefits subsequent to that, namely that the industry gets a whole load of new trained PlayStation programmers who are moving into the third-party community or even working for us.

NG: Do you realistically expect that the next *Boulderdash*, or *Doom* will come out of a *Yarouze*-equipped garage?

Phil: Well, "Why not?" is, I guess, the answer to that! If you look at some of the seminal games of all time they are always one-man enterprises. From Peter Molyneux sitting on his own developing *Populous* to Will Wright's *Sim City* or Sid Meier's *Civilization*, you know these were all one-man enterprises. So why shouldn't the next great game be developed in this way?

NG: And of course Sony will have exclusive rights to *Yarouze* developed games.

Phil: Well, not necessarily. We can't deny that that's an opportunity for us. But similarly there is opportunity for the third-party software companies to start working with these people as well. And yet, if we see the next *Sim City* or the next *Populous*, we're going to be falling over ourselves to work with these people and to help make the game into a fully fledged commercial product. But that's not the only reason why we are doing this.

There is a whole other business model out there for people to make money from perhaps getting involved with a compilation disk that we would do as a "best of" from this program. I guess you could also call *Yarouze* job training — so people can learn enough to go get a job at some hot start-up company and make their millions on the stock market ten years from now.

NG: Just how big do you estimate the demand for *Yarouze* to be in unit sales?

Phil: No idea, nor does it matter. I mean the infrastructure to support one user or twenty thousand is exactly the same. This is not PlayStation 2. This is not a mass market consumer proposition. But it is a highly strategic initiative that reaps the reward three years down the line. It does not really matter how big it is.

NG: What's the best case scenario for *Yarouze*?

Phil: I think the best thing would be a situation where **Next Generation** reviews a game. You give it a fantastic score and in interviewing the programmer, he confesses that this game actually started out when he was a member of the *Yarouze* development community and he took it to xyz company and they said, "This is great. Here's a job. Come work as part of a team and build your masterpiece into a fully fledged consumer product." That would be the dream.

I think that there is another benefit or another dream which would be that there are legions of programmers coming into the marketplace creatively and technically trained on PlayStation. Obviously, it benefits our third-party program for many many years to come. So it's kind of like insurance in that way.

In a perverse way, I want to see code listings in **Next Generation**. Typing in stuff from magazines. That's the same concept. I got my start in videogames by typing in listing of games from magazines. I learned to program by screwing around with them and saying, "This doesn't work, but what about this?" It would be great to see code examples in little routines printed in magazines. That would be great.

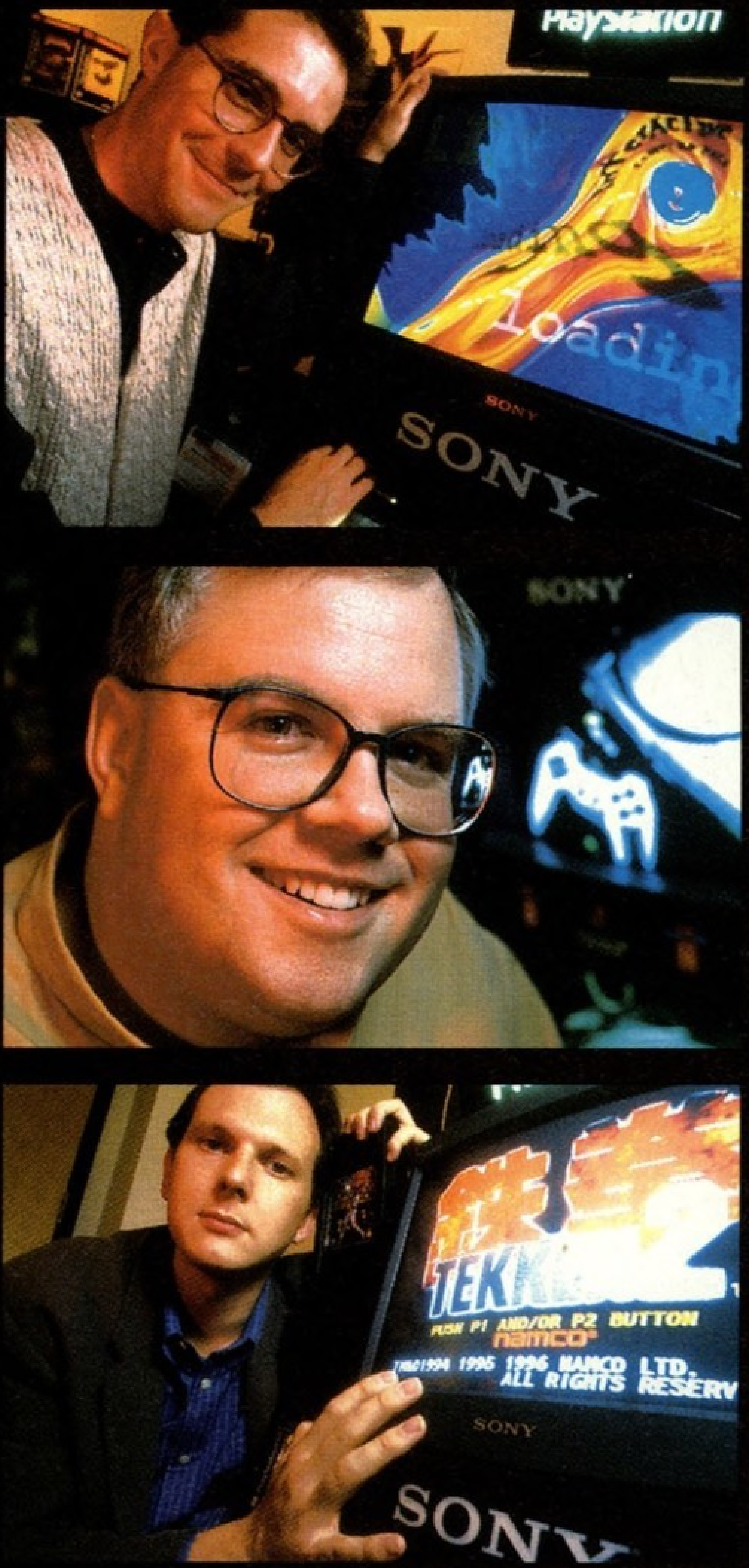
NG: Why call it *Yarouze*?

Andrew: It's a Japanese slang expression for "Let's do it!" It was a tag-line used for just about everything in the early PlayStation marketing — "1,000,000 units? *Yarouze*!"

Phil: I think it embodies the creative empowerment that this program is all about. Which is: "OK, Mister Gameplayer. You have always sat at home and said, 'I could do better than that.' You know, 'This game sucks, I know I could do better than that.' Well here you are, go do it. *Yarouze*!"

NG: Will SCEA be pursuing education opportunities with *Yarouze*?

Phil: Oh yes, absolutely. A logical outlet for this product is into computer graphics and computer science courses at varying educational levels, be it the Master's Degree level or moving down to vocational training and technical colleges. We'll have a mechanism for government and educational establishments to buy systems.


PlayStation
loading
SONY
SONY
TEKKEN 2
PUSH P1 AND/OR P2 BUTTON
namco
©1994 1995 1996 NAMCO LTD. ALL RIGHTS RESERV
SONY

From top to bottom, Sony's Andrew House, Bill Rehbock, and Phil Harrison

NG: How much will *Yarouze* change the impression of PlayStation in the eyes of the man on the street? Perhaps someone who has no intention of ever actually buying one?

Andrew: I think that it will certainly enhance the overall brand image of PlayStation. I think that it lends an air of creativity to the platform. It obviously shows that we are forward-looking and that we are setting up plans for the future. I think there is a lot of positive impact to be had.

I also think that there will be significant interest from general consumers as to what's developed there and I think that you create a very positive spiral because you have people who are creating their own stuff and want other people to look at it and you'll have a significant number of interested consumers who will be very keen to seen how it's emerging.

Phil: I think that knowing that this particular game was created by "one of us" has some appeal. Empowerment is a very compelling hook for our target consumer. And so knowing that you can do this with a PlayStation, even if you yourself are not going to it or don't have the skills or the inclination, just knowing that PlayStation is capable of this is a very compelling part of the brand ethos, the thing that makes PlayStation what it is. And I think that will become stronger as time goes on.

Bill: The other thing that you are going to see is that people will be seeing the realization of their wishes. If a guy is up on a web site saying, "Boy it would be really cool if a game like X existed," who's more likely to pick up on it and do it, Virgin or a *Yarouze* member? No *Yarouze* member is going to say, "OK we'll get a focus group to check it out, and run it past the midwest sales manager." They'll just do it and upload it to the website.

NG: How do you expect the distribution and creation model to work?

Phil: I think that what we are expecting to see is that along with the executable program, the source code goes up as well. And a lot of people will say, "This is how I did it." Then I expect two or three people to band together and say, "Let's work on a masterpiece. You do this section of the game, and we will do this bit, and my mate down at USC is going to do the graphics."

Bill: One thing that we have seen in Japan is that we have *Yarouze* members who are not programmers or artists, but are designers who have good ideas. They get together with the programmers and artists and say, "Here is this diamond in the rough, but it'd be really cool if you did this and this and this."

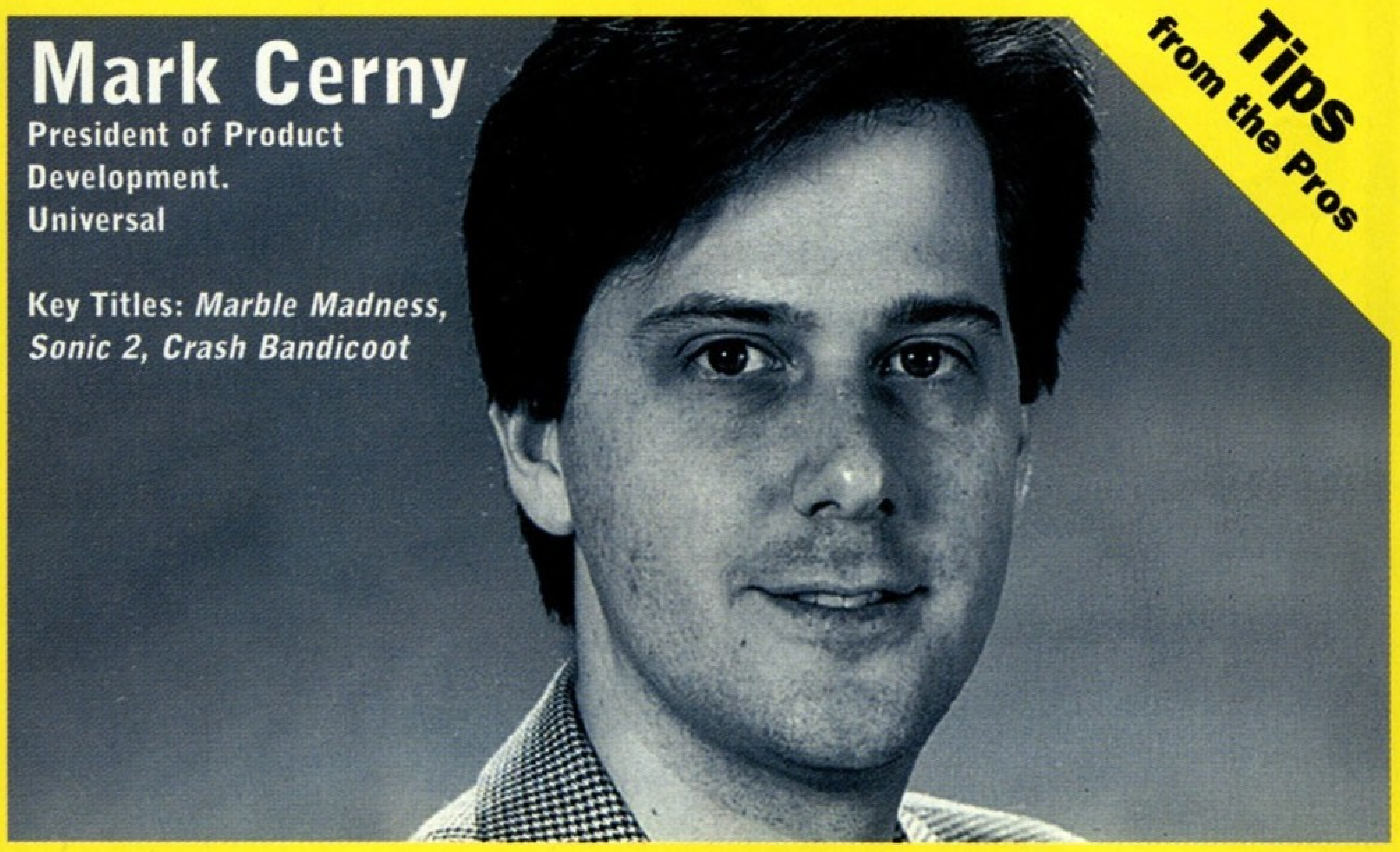
Then someone else says, "You know that programmer art really sucks," and contributes art to it. So we have these titles that are in incubation, that have scattered development members and wow — it's 1975 all over again.

Phil: *Yarouze* is almost the perfect Internet application in that respect. You can work with people you've never met in different countries, in different time zones, where you never have any real-time interaction with them. I think that's very much the kind of collective ideal of this, and it's something which is completely out of our control.

And that's the appeal of it! There're no content guidelines, there's no approval process, there is no requirement that the product fulfill certain taste or commercial requirements. Because this is not a retail program.

Full products going onto shelves are sensitive to the dynamics of the marketplace, because there're only so many linear feet of shelf space in stores. But with this, the shelf space is infinite. It's virtual shelf space. So people can go and do what they like, and they don't even have to be games. They don't just have to be for a typical target demographic. They can be all kinds of weird and wonderful things, and that's part of the appeal — what are these people going to come up with?

NG: What about the limited RAM? Is 2MB enough to create a great title in?



Mark Cerny

President of Product Development. Universal

Key Titles: ***Marble Madness, Sonic 2, Crash Bandicoot***

Tips from the Pros

Cerny completed *Marble Madness* when he was only 17. Think about that...

Ten tips for making money from your *Yarouze* game

> **"What is the videogame equivalent of the black and white art movie? We're going to get to see."**

1) From time to time, invite PlayStation owners over to play your game
There are many things that can be learned by watching them play, the biggest being if they are having fun!

2) Keep the rules of the game simple
Ideally, first-time players should understand and enjoy the game without instructions.

3) Make sure the player doesn't get "stuck in a rut"
If the difficulty progression of the game is not smooth, players will become stuck at a certain point (again, inviting PlayStation owners over to play the game will help).

4) If the game is a genre game, pay attention to the unwritten rules of the genre
For example, while there may not be abstract reasons why a certain controller scheme is best, there are definitely conventions. If you are doing an action game with a jump, survey the successful games in the genre to see if there is a button that is always used.

5) Use assembly language to speed your frame rate
This will help you compete with other games in the category.

6) Put in music, even if it's temporary
This will help even a prototype game feel alive!

7) Take it as far as you can before you start to look for a publisher
The farther a game is to towards completion, the better your negotiating position.

8) Realize your (probably) can't take it all the way
For example, professional sound effects and music costing $50,000 will make the final product much more slick, but you won't have the resources to do this until you've lined up a production deal. The most important thing to complete is the gameplay.

9) Pick your publisher carefully
Is this a company that puts out game you think are good? Is it stable?

10) Bet on your success
If it is possible to forgo some money early on, in return for big money later on, do it!

Phil: Well it's 3.5 if you include the 1MB of VRAM and half a meg of sound RAM. So that's a lot if you think that *Ridge Racer* is the benchmark. Most users don't have a CD-ROM burner and an SGI to create 15-minute rendered intros, so we don't think it's a problem. It also puts the focus on the essence of the idea rather than the production. On commercial software, where you're charging fifty bucks for the product, the focus is on the production values and production quality to justify the price. This doesn't have those kind of commercial dynamics applied to it. And so you can have a really really simple idea that would never stack up on a store shelf. But it is an absolutely boiled down refined well-formed concept. One of the things we will do is look at these pure concepts and evaluate them and go, "Wow that would make a fantastic product" and work with the creator to turn it into something great.

"...establishing a new standard in the platform genre."
-Next Generation

"Massive environment, go anywhere gameplay and amazing depth give Bubsy an entirely new perspective."
--DieHard GameFan

"BUBSY 3D looks to set the new standard in action platform gaming." --PSExtreme

PlayStation™

SEGA SATURN™

RATING PENDING
RP
CONTENT RATED BY ESRB

wea
Visual Entertainment

ACCOLADE™

HIGHTAIL IT INTO A NEW DIMENSION!

http://www.accolade.com Call 1-800-245-7744 for product information

BUBSY® AND THE BUBSY CHARACTERS ARE REGISTERED TRADEMARKS OF ACCOLADE. DISTRIBUTED BY WEA (WARNER ELEKTRA ATLANTIC CORP.), A WARNER MUSIC GROUP COMPANY. © 1996 ALL RIGHTS RESERVED. LICENSED FOR SONY COMPUTER ENTERTAINMENT AMERICA FOR USE WITH THE PLAYSTATION GAME CONSOLE. PLAYSTATION AND THE PLAYSTATION LOGOS ARE TRADEMARKS OF SONY COMPUTER ENTERTAINMENT INC. © 1996 ALL RIGHTS RESERVED. LICENSED FOR SEGA OF AMERICA FOR USE WITH THE SEGA SATURN GAME CONSOLE. SEGA SATURN IS A TRADEMARK OF SEGA OF AMERICA, INC. © 1996 ALL RIGHTS RESERVED. THE RATING ICON IS A REGISTERED TRADEMARK OF THE INTERACTIVE DIGITAL SOFTWARE ASSOCIATION.

imagine
a new way of publishing
Next Generation OnLine, http://www.next-generation.com/